

THE NEW EMIT

A. U. Luccio

June 1990

Collider Accelerator Department
Brookhaven National Laboratory

U.S. Department of Energy
USDOE Office of Science (SC)

Notice: This technical note has been authored by employees of Brookhaven Science Associates, LLC under Contract No.DE-AC02-76CH00016 with the U.S. Department of Energy. The publisher by accepting the technical note for publication acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this technical note, or allow others to do so, for United States Government purposes.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

THE NEW EMIT

BOOSTER TECHNICAL NOTE
NO. 172

A. LUCCIO

JUNE 26, 1990

ALTERNATING GRADIENT SYNCHROTRON DEPARTMENT
BROOKHAVEN NATIONAL LABORATORY
UPTON, NEW YORK 11973

THE NEW EMIT

A.LUCCIO

4 June 1990

1. INTRODUCTION

An important task during the commissioning and operation of the AGS booster will be the measurement of the emittance of the injected beam. For this, a computer code (EMIT) was developed in the past by D. Weygand of the AGS¹, based on the formalism proposed by K. Ebihara et al.² We have revisited this code, changing some of its features and broken it in several modules that perform the various tasks for emittance measurement. The model MAD³ is used both to calculate the transfer matrices and to build a simulated beam used to check the correct operation of the algorithms. The overall flow of the program is controlled by AEGIS scripts invoked from the Booster menu. An example of application to the LTB (linac to booster) proton injection line is shown.

2. THE ALGORITHMS

2.1. Principles of emittance measurements

The emittance ellipses in a given point of a beam line are described by the equation

$$\gamma x^2 + 2\alpha x x' + \beta x'^2 = \epsilon \quad (1)$$

with α , β , and γ Twiss parameters and ϵ a value for the percent emittance, corresponding to a given beam relative intensity, and x a transverse coordinate, horizontal or vertical. The coordinates x and x' transform along the line from a point "0" to a point "m" according to

¹ D. P. Weygand. "EMIT", ACS/DOC 24 May 1988, Rev> 28 June 1988

² K. Ebihara, M. Tejima, T. Kawakubo, S. Takano, Z. Igarashi and H. Ishimaru, "Non-destructive emittance measurements of a beam transport line", Nucl. Inst. Methods **202** (1982) 403-409

³ F. Ch. Iselin, and J. Niederer, "The MAD Program (Methodical Accelerator Design, Version 6, User's Reference Manual", Rept. CERN/LEP-TH/87-33, Geneva, April 1987

$$\begin{pmatrix} x \\ x' \end{pmatrix}_m = \begin{pmatrix} C & S \\ C' & S' \end{pmatrix} \begin{pmatrix} x \\ x' \end{pmatrix}_0 \quad (2)$$

where the four quantities C, S, C' and S' are known, once the geometry and the excitation of each magnet is specified. Only three coefficients are independent since it is

$$CS' - SC' = 1 \quad (3)$$

The transverse size of the beam in the point "m" is given by

$$w_m = \sqrt{\beta_m \epsilon} \quad (4)$$

The coefficients of the ellipses (1) transform along the line, from point "0" to point "m" as

$$\begin{pmatrix} \beta \\ \alpha \\ \gamma \end{pmatrix}_m = \mathfrak{R} \begin{pmatrix} \beta \\ \alpha \\ \gamma \end{pmatrix}_0 \quad (5)$$

where \mathfrak{R} is the transfer matrix⁴

$$\mathfrak{R} = \begin{pmatrix} C^2 & -2CS & S^2 \\ -CC' & CS' + SC' & -SS' \\ C'^2 & -2C'S' & S'^2 \end{pmatrix} \quad (6)$$

In particular, for the β function it is

$$\beta_m = \frac{w_m^2}{\epsilon} = C^2 \beta_0 - 2CS \alpha_0 + S^2 \gamma_0 \quad (7)$$

The elements of the transfer matrix of Eq. (2) and therefore of \mathfrak{R} can be easily calculated with an optical code like MAD, since they can be expressed in terms of the twiss functions and the phase advance ϕ as follows

⁴ e.g. see: K.Steffen, "Basic course on accelerator optics" in CAS Cern Accelerator School, CERN 85-19, 27 November 1985, p. 45

$$\begin{aligned}
C &= \sqrt{\frac{\beta_m}{\beta_0}} (\cos(\phi_m - \phi_0) + \alpha_0 \sin(\phi_m - \phi_0)) \\
S &= \sqrt{\beta_m \beta_0} \sin(\phi_m - \phi_0) \\
C' &= -\frac{1}{\sqrt{\beta_m \beta_0}} ((\alpha_m - \alpha_0) \cos(\phi_m - \phi_0) + (1 + \alpha_m \alpha_0) \sin(\phi_m - \phi_0)) \\
S' &= \sqrt{\frac{\beta_0}{\beta_m}} (\cos(\phi_m - \phi_0) - \alpha_m \sin(\phi_m - \phi_0))
\end{aligned} \tag{8}$$

with any reasonable choice of the twiss parameters at the "0" point. Once calculated, the matrices will still remain valid for any values of the twiss parameters.

So, with MAD, and a series of k-values for the quadrupoles in the line, the elements of \mathfrak{R} are calculated and stored in a file. For each setting, values of the beam widths w_m at various beam intensity percent are measured at a measuring point "m" and from Eq.(7) the values of $\epsilon\beta_0$, $\epsilon\alpha_0$, $\epsilon\gamma_0$ can be calculated. For the three quantities, we need at least three measurements in the same spot, with three different magnet settings, or equivalently, three measurements at different positions with the same setting, or a combination of both. More than three measurements will improve the statistics.

Finally, since it is

$$\alpha_0^2 = \beta_0 \gamma_0 - 1 \tag{9}$$

the emittance is obtained as

$$\epsilon = \sqrt{(\epsilon\beta_0)(\epsilon\gamma_0) - (\epsilon\alpha_0)^2} \tag{10}$$

2.2. Beam width. smoothing of data.

2.2.1. FFT smoothing.

Beam profiles are read on multiwire monitors, or "harp". The profile of the beam is found from the read values. In order to find the beam widths w_m [Eq.(4)] for various emittance percents, criteria are desirable to smooth and fit the data and to eventually discard sets of data. A first smoothing is optionally done in order to suppress some noise due to the finite number of wires. The algorithm used is a Fast Fourier Transform: the FFT spectra of the readings is taken by a forward transform, the highest harmonics are eliminated and then the spectrum is back transformed. It is found that the needed information is contained in the first four to six harmonics.

An example of FFT smoothing is shown in Fig.1. Simulated "readings" from an harp with 32 wires are processed and therefore 16 FFT amplitudes are available in frequency space. The figure shows what happens when only 12, 8, 6 and 4 harmonics are respectively kept.

2.2.2. Newton-Raphson fit.

The next step is data fit. If we want to represent a set of data, $f_i \equiv f(x_i)$ by a functional of a pre-established form $g(p, x)$, where p represents a set of parameters, a good criterion is the least squares method

$$S(p) = \sum_i [f_i - g(p, x_i)]^2 = \min \quad (11)$$

The problem reduces to find the values of the parameters $\{p\}$ for which the partial derivatives of S vanish:

$$F_j = \sum_i [f_i - g(p, x_i)] \frac{\partial g}{\partial p_j} = 0 \quad (12)$$

The F_j can be considered the components of a vector in a space with the same number of dimensions than parameters in the definition of the functional g . To approximatively solve the system (12) is equivalent to minimize the norm of this vector \mathbf{F} . The fastest algorithm is an extension to a multi dimensional space of the familiar Newton-Raphson method used to find the zeros of a function of a real variable; only here, in place of the local derivative of

the function, the Jacobian matrix containing the partial derivatives of the components of \mathbf{F} with respect to the variables $\{\mathbf{p}\}$ will be used⁵

$$J_{jk} = \frac{\partial F_j}{\partial p_k} = \sum_i \left\{ [f_i - g(\mathbf{p}, x_i)] \frac{\partial^2 g}{\partial p_j \partial p_k} - \frac{\partial g}{\partial p_j} \frac{\partial g}{\partial p_k} \right\} \quad (13)$$

After a starting value for the parameters is assumed: $\{\mathbf{p}_0\}$, the Newton-Raphson algorithm finds the values of $\{\mathbf{p}\}$ at iteration $n+1$ from the values at iteration n as follows

$$\left(\frac{\partial F_j}{\partial p_k} \right)^{(n)} (p_k^{(n)} - p_k^{(n+1)}) = F_j^{(n)} \quad (14)$$

If the procedure converges, the norm of \mathbf{F} decreases. When a satisfactorily small value is reached, the iteration stops and the final values of the parameters are taken.

Let us remember that the Newton-Raphson algorithm may converge in an oscillatory mode, in a monotonic mode, or diverge. Moreover, if it converges, it may converge to a nonphysical value, therefore great care has to be exercised and the flow of the algorithm must be checked. The choice of a starting point is very important. Our code tries to address these points; for example, if the convergence is not achieved after a few iterations, if the algorithm shows a diverging behaviour, or if some of the parameters start to slip in a some nonphysical region, a different starting point is chosen and the iteration repeated. There will be a brief discussion of this in the examples below.

2.2.3. Example of functional: Gaussian

If the functional form of the fitting function is a gaussian of amplitude a and half-width σ , centered on x_0 (therefore, three parameters)

⁵ J.M.Ortega and W.C.Rheinboldt, "Iterative solution of nonlinear equations in several variables", Academic Press 1970, p. 181

$$g(x_i) = a \cdot e^{-\frac{(x_i - x_o)^2}{2\sigma^2}} \quad (15)$$

the Jacobian and the vector \mathbf{F} are, respectively

$$\begin{pmatrix} \frac{1}{a} \sum P_i & \frac{1}{\sigma^2} \sum u_i P_i & \frac{1}{\sigma^3} \sum u_i^2 P_i \\ \frac{1}{a} \sum u_i P_i & \frac{1}{\sigma^2} \sum u_i^2 P_i - \sum Q_i & \frac{1}{\sigma^3} \sum u_i^3 P_i \\ \frac{1}{a} \sum u_i^2 P_i & \frac{1}{\sigma^2} \sum u_i^3 P_i - 2 \sum u_i Q_i & \frac{1}{\sigma^3} \sum u_i^4 P_i \end{pmatrix} \begin{pmatrix} \sum P_i \\ \sum u_i P_i \\ \sum u_i^2 P_i \end{pmatrix} \quad (16)$$

with

$$u_i = x_i - x_o, \quad P_i = g(x_i)(f_i - 2g(x_i)), \quad Q_i = g(x_i)(f_i - g(x_i)) \quad (17)$$

Fig. 2 (a, b) show examples of gaussian fit of two sets of data. These were old data processed by EMIT and were taken as working examples. Criterion to stop the NR iteration was

$$|\mathbf{F}| \leq 0.001 \quad (18)$$

(where the norm of \mathbf{F} is of the order of one to start with).

Fig. 2a shows a set of "nice" data. Convergence was achieved in three iterations both in the horizontal and vertical plane. The rms relative error in the amplitude resulted in 5 % and 2 %, respectively over the whole range of values. The error is somewhat larger if we only consider the central part of the distribution, where values are significantly different from zero. The residual error is also plotted in the figure. Fig. 2b shows much worse data. In the upper half of the figure (horizontal) there were very few significant data, in the lower half (vertical) the data showed a double peaked distribution. In both cases convergence was achieved in two steps, after the starting point was automatically readjusted following an unsuccessful attempt towards convergence

2.2.4. Example of functional: raised cosine.

The functional is

$$\begin{cases} g = \frac{a}{2} \left(1 + \cos \left(\pi \frac{x - x_0}{2\sigma} \right) \right) & ; \quad |x - x_0| \leq 2\sigma \\ g = 0 & \text{elsewhere} \end{cases} \quad (19)$$

Applications to the same data of Fig. 2 are shown in Fig. 3 (a,b). As for the gaussian, the data of Fig. 3a were processed with few iterations, while the data of Fig. 3b required a readjustement of the starting point. Accuracy for the gaussian and the raised cosine for the examples at hand are comparable.

2.3. Simulation

To check the correct functioning of the algorithms, we used the model MAD to simulate an injected beam. First, a cloud of particles representing the beam at the entrance to the line was created by random number generation, within an ellipse defined by the 95% beam emittance, both in the horizontal and vertical phase spaces. Then, the emittance figure was propagated to the location of the harps and "readings" were taken at the wire locations, for a variety of settings of the quadrupoles preceding the given harp. Thus, we created a file of simulated readings to play on.

The procedure for particle generation was the following: we arbitrarily decided that the beam was gaussian, with particles randomly distributed in a number of bins in transverse position, between $-\sqrt{\beta\epsilon}$ and $\sqrt{\beta\epsilon}$ and in angle between $-\sqrt{\epsilon/\beta}$ and $\sqrt{\epsilon/\beta}$. We kept only those particles falling within the contour defined by Eq. (1), and we wanted the total number of particles in each plane, within the ellipse, to be small enough to make the distribution somewhat noisy.

We assumed a reasonable value for the emittance (for the LTB line: 20.4 mm-mrad in horizontal, and 37.9 mm-mrad in vertical), and 32 bins for the statistical distribution. In total, 160 starting particles were used in the simulation, as shown in Fig.4. The phase space coordinates of the particles were extracted at random, in a way that their density distribution in the transverse coordinate was gaussian.

The precalculated values of C , S , C' and S' of Eq. (8) are used to transfer the generated particles to the harps, according to Eq. (2). At the harps, particles are binned at the 32 wire locations and stored as "readings". Fig. 5 shows examples of plots of simulated readings in the LTB for a particular setting of the quads.

3. FLOWCHART of NEW EMIT

NEW EMIT consists of high level (mostly FORTRAN) code pieces arranged through Apollo AEGIS scripts. Its flowchart is shown in Fig. 6. In the description of the flowchart we will explicitly make reference to the LTB line, as an example of application. The layout of the LTB is shown in Fig. 7 produced from the database as described in⁶.

The flow in NEW_EMIT is controlled by a series of AEGIS scripts, reproduced in the appendices. The main script is called E0 and is given in Appendix A.

3.1. Entry point A.

With ref. to Fig. 6, the first task is to prepare a file containing the transfer matrices between the point "0" of the line where the emittance is sought and selected monitors. By default, point "0" is the entrance to the line. The geometry of the line and some relevant parameters, like the "usable flag" that tells us whether a given element is in good health, are contained in the database: use of the program IMPORT⁶ creates a file "machine_element_parameters" with the needed data.

The AEGIS script, EMAD in Fig. 6, reproduced in Appendix B, invokes MAD many times to create matrices for a set of pre-established quadrupole k -values. The script QSET (Appendix C) contains the string of values to be used, together with a string of the harp names and of the quadrupole names (all harps and quads in the line). This script is customized to a given line: in principle, there is one for each line in the booster complex. Each k -string contains a series of values, typically three, corresponding to the standard k for the given quad, and that k plus or minus 20%. These values can be edited.

⁶ E.H.Auerbach and A.Luccio, "Programming interface with the booster database. Examples", BNL Report Booster Tech. Note # 162, April 6, 1990

There is a reference input file to MAD: "mad.in.0", previously created by the program "madwrite" by E.H.Auerbach. This input is automatically scanned by program "mad_tune" to find all the quads preceding a given monitor in the line. If the elements are correctly found and the "usable flag" tells us that they can be used, a file "mad.in" is created for the first value in the k-string and MAD is run for some standard values of the Twiss parameters at the beginning of the line. With the option "TWISS", MAD writes an output file that is read to find betas and phase advance at the "m" points; with this, the transfer matrices are calculated according to Eqs. (8), (2) and (6) by program "ematrix" and written to the file "qm.data" together with informations on the relevant k set, and harp name. The process is repeated with the next element of the m-, q- and k-strings, until the file "qm.data" is completed. Using an Apollo Domain 4000 workstation, this preparatory process took about 12 minutes for 36 cases.

A fragment of qm.data for the LTB line is shown in Appendix F.

3.2. Entry point B.

During booster commissioning, the first set of quadrupole k-values will be converted to currents in the magnets ("machine"), the beam injected and the harps, or profile monitors, read. This is shown in the flowchart: the sets of quad k-values are sent to the "machine". The machine is supposed to react and produce the harp wire readings that are written to the file "wire.data".

(In the LTB line each harp has 32 horizontal wires and 32 vertical wires; spacing between wires is 0.75 mm, spanning a total width of 23.25 mm).

In order to simulate the process of data acquisition and emittance calculation and test the software, and to have a permanent model checking capability, we want to create a mock injected beam and harp "readings". This is done in the script EMODEL, reproduced in Appendix D, following the procedure described in Sec. 2.3. EMODEL can be optionally run from E0 (booster or model?).

To create a beam, informations on the injection Twiss parameters are taken from the database- derived file machine_element_parameters and a reasonable value for the emittance is assumed. This is done by the program "emock".

3.3. EFIT

The script EFIT, reproduced in Appendix E, first processes wire.data, created either by the real machine or by the model, with FFT smoothing (program "efft") and least squares-Newton-Raphson (program "nr") to find an acceptable fit. Criteria are built herein to decide whether a given functional is a good representation. This analysis is somewhat subjective, since by successive iterations we may fit almost anything to everything. Typically, if a convergence iteration fails after readjusting the starting point, we may want to discard the set of data altogether. We may also do so if the residual error in the fitted amplitude is more than, say, 10 % of the original amplitude.

A further programming module, "emittance", reads the smoothed and fitted data and does two things: (i) calculates the beam widths for different beam intensity percentages (typed in from terminal) and (ii) using the transfer matrices from qm.data, and performing some statistical analysis (assuming that the data are redundant), produces the sought emittances at the entrance point. For the statistics, a number of combinations are taken among the available datasets.

The calculated emittances are finally appended to the file "emit.data", that contains also the harp readings. Plot files to display the calculated emittances are also produced for CPLOT⁷. Emittances may be stored back in the database using EXPORT⁶.

4. EXAMPLES of mock beam readings and emittance calculation for the LTB. .

Representative points of the mock beam of Fig. 4 were transferred to a harp and the intensity "read" on 32 wires with the proper spacing, horizontally and vertically. This was repeated for various settings of the quadrupoles in the line upstream of the given monitor.

Fig. 8 (a,b) show typical reading at MW035 when QH1 and QV2 are varied, respectively. The effect of the horizontal quadrupole QH1 is seen in (a), for $k = 1.2, 1.4, 1.6$ (standard value is 1.4). The 32 points have been connected by a broken line for better readability.

⁷ A. Luccio. "CPLOT: an Apollo plotting program using CALCOMP and GPR", AGS booster Tech. Note # 156, January 9, 1990.

The effect of the vertical quadrupole QV2 is seen in (b) for $k = -1.4, -1.6, -1.8$ (standard value -1.6).

Fig. 9 (a,b) show the "readings" at MW107 when varying QH8 (0.3, 0.5, 0.7) and QV9 (-0.4, -0.6, -0.8) respectively.

APPENDIX G shows a fragment of emit.data completed with the calculation of the 50% emittance for the example above.

Appendix A. E0: main script to run the package NEW_EMIT

//acn40d01/reality/luccio/pline/harp/E0

06/20/90 3:21 PM p.1

```
#####
#                                     E0
#
# //acn40d01/reality/luccio/pline/harp
# A.Luccio, BNL Mar 1990
# Mod: Thursday, May 7, 1990 11:49:57 am (EDT)
#####
eon
read -p "run # " -type int run
verbose := false
read -p "verbose [y/n]? " v ; if ((^v = y)) then verbose := true endif
#-----
# initialize files

if existf emit.data then dlf emit.data endif
args ^'date' "file emit.data" "run ^run" @
"result of wire data analysis" >> emit.data
#-----
# build transfer matrices with mad
# script QSET contains quad settings,
# returns with m_string, q_string, k_string
args "==== QSET ====="

        c := y
        t := n
if ((^verbose)) then read -p "re-calculate matrices [y/n]? " t endif
if (( ^t = y )) then

if existf qm.data then dlf qm.data endif
args ^'date' "file qm.data" "run ^run" @
"quadrupole setpoints and transfer matrices between 0 and harps" @
"from: machine_element_parameters" >> qm.data

        set := 1
        mon := 1
        q := 1
        source QSET ^mon ^q
for monitor in ^m_string by word
        source QSET ^mon ^q
        for quad in ^q_string by word
        source QSET ^mon ^q
                for k_value in ^k_string by word
                        args "=== EMAD: dataset ^set, k1(^quad) = ^k_value"

                        EMAD ^run ^set ^monitor ^quad ^k_value
                        set := ^set +1
                endfor
                q := ^q +1
        endfor
        mon := ^mon +1
endfor

# qset and matrices are written to qm
ld -dtm qm.data
cpf qm.data data/qm.data -r
```

Appendix A.

```

read -p "continue [y/n]? " c
endif
#(typically 12 min for 36 cases, LTB, May 10,1990)
#-----
if (( ^c = n )) then EQUIT endif
#-----
# Simulation. Prepare mock beam

read -p "booster or model [b/m]? " bom
if (( ^bom = m )) then
  read -p "prepare mock beam [y/n]? " mock
  if (( ^mock = y )) then
    EMODEL

  endif
endif
#-----
# Read wire_data. Smooth. Find widths. Plot

s1 := 1
s2 := 10
if ((^verbose)) then read -p "dataset from - to - " -type int s1 s2 endif
for set := ^s1 to ^s2
  args " " "dataset ^set" >> emit.data

# send quad set values to machine: qset.data --> machine
# obtain harp data          : machine --> wire.data.^set

  if ((^bom = b )) then
    if existf data/wire.data.^set then
      cpf data/wire.data.^set wire.data -r
    else exit
    endif
  endif
  if ((^bom = m )) then
    if existf data/model_wire.data.^set then
      cpf data/model_wire.data.^set wire.data -r
    else exit
    endif
  endif

  args "===== EFIT ====="

  EFIT ^run ^set ^verbose
endfor

#(typically 6:30 min for 10 cases, LTB, May 10,1990)
EQUIT
#####

#####
#
#          EQUIT
#####
#close files. List files. Quit
#
cpf emit.data data/emit.data.^run -r
ld -dtm qm.data data/wire.data?* data/emit.data?*
xdmc dq
#####

```


Appendix B. EMAD and MRUN: scripts to run MAD

//acn40d01/reality/luccio/pline/harp/EMAD

```
#####
#                                     EMAD
#
# //acn40d01/reality/luccio/pline/harp
# A.Luccio, BNL Mar 1990
# Mod: Thursday, May 7, 1990 11:49:57 am (EDT)
#####
# ^1=run, ^2=dataset, ^3=monitor_name, ^4=quad_name, ^5=k1(quad)
#####
eon
  args " " "dataset ^2" >>qm.data

cpf data/mad.in.0 mad.in -r

mad_tune<<%
^4
^5
mad.in
mad.in.temp
%

cpf mad.in.temp mad.in -r
#xdmc ce mad.in ; read -p "continue [y/n]? " c

MRUN

# read mad.twiss, prepare transfer matrices, write file qm.temp

ematrix <<%
data/machine_element_parameters
^3
mad.twiss
qm.temp
%

catf qm.temp >>qm.data ; dlf qm.temp
#####
```

//acn40d01/reality/luccio/pline/harp/MRUN

```
#####
#                                     MRUN
#
# //acn40d01/reality/luccio/database
# A.Luccio, BNL, March 1990
# Mod: Friday, April 6, 1990 5:16:38 pm (EDT)
#####
args "=== run mad ==="
if existf TWISS then dlf TWISS endif
if existf mad.out then dlf mad.out endif
/reality/luccio/model/mad.run7
if existf TWISS then mvf TWISS mad.twiss -r
else args "file TWISS does not exist!" ; catf mad.out
endif
#####
```

Appendix C. QSET: script containing the m-, q- and k-string (LTB line).

```
//acn40d01/reality/luccio/pline/harp/QSET

#####
#
#                               QSET
#
# //acn40d01/reality/luccio/pline/harp
# A.Luccio, BNL Mar 1990
# Mod: Thursday, May 10, 1990 11:49:57 am (EDT)
#####
# ^1=quad_name, ^2=k1(quad)
#####
eon
      m_string := "BLI.MW035 BLI.MW107"

select ^1
case 1 ; q_string := "QH1: QV2: QH3: QV4: QH5:"
case 2 ; q_string := "QH6: QH7: QH8: QV9: QH10: QV11: QH12:"
endselect

select ^2
case 0 ; k_string := "1.4 -1.6 .6 -1.2 1.6 1.466 .8 1.107 -1.165 .87 -.813 1.6"
case 1 ; k_string := "1.2 1.4 1.6"
case 2 ; k_string := "-1.4 -1.6 -1.8"
case 3 ; k_string := "1.4 1.6 1.8"
case 4 ; k_string := "-1.5 -1.7 -1.9"
case 5 ; k_string := "1.5 1.7 1.9"
case 6 ; k_string := "0.4 0.6 0.8"
case 7 ; k_string := "0.5 0.7 0.9"
case 8 ; k_string := "0.3 0.5 0.7"
case 9 ; k_string := "-0.4 -0.6 -0.8"
case 10 ; k_string := "0.4 0.6 0.8"
case 11 ; k_string := "-0.5 -0.7 -0.9"
case 12 ; k_string := "1.2 1.4 1.6"
endselect

args
#####
```

Appendix D. EMODEL: script to create a simulated beam and harp redings.

```
#####
#
#                               EMODEL
#
# //acn40d01/reality/luccio/pline/harp
# A.Luccio, BNL May 1990
# create mock beam for emittance measurements simulation
# Mod: Thursday, May 7, 1990 11:49:57 am (EDT)
#####
eon
if existf data/ellix_0.pdat then dlf data/ellix_0.pdat -l endif
args "! ^'date'" >> data/ellix_0.pdat
if existf data/elliy_0.pdat then dlf data/elliy_0.pdat -l endif
args "! ^'date'" >> data/elliy_0.pdat
if existf data/ellix_m.pdat then dlf data/ellix_m.pdat -l endif
args "! ^'date'" >> data/ellix_m.pdat
if existf data/elliy_m.pdat then dlf data/elliy_m.pdat -l endif
args "! ^'date'" >> data/elliy_m.pdat

emock <<%
32,19072452
ellix_0.temp
elliy_0.temp
qm.data
ellix_m.temp
elliy_m.temp
%

#xdmc dq

catf ellix_0.temp >>data/ellix_0.pdat ; catf elliy_0.temp >>data/elliy_0.pdat
catf ellix_m.temp >>data/ellix_m.pdat ; catf elliy_m.temp >>data/elliy_m.pdat

/reality/luccio/graf/data_bias <<%
ellix_0.temp
1
1
0.01,0.01,1,1
0
ellix.pbias
%

/reality/luccio/graf/data_bias <<%
elliy_0.temp
1
1
-0.01,-0.01,1,1
0
elliy.pbias
%

p := y
while (( ^p = y )) do
#/reality/luccio/graf/PLOT data/elli.ppar data/elli_prof.pin 0 1
/reality/luccio/graf/PLOT data/elli.ppar data/elli.pin 0 1
read -p "repeat plot [y/n]? " p ; if (( ^p = n )) then exit endif
enddo
dlf temp ellix_0.temp elliy_0.temp cplot.in ellix.pbias elliy.pbias

# -----
# return to main

E0
#####
```

Appendix E. EFFT: script to fit data and calculate emittances.

```
#-----
# FFT smoothing + plot

read -p " FFT smoothing [y/n]? " smooth
if (( ^smooth = y )) then
read -p "fft filter = " -type int filter
args "===== EFFT ====="

efft <<%
^filter
wire.data
wire.temp
ex.pdat
ey.pdat
%

mvf wire.temp wire.data -r

EFPLOT ^1 ^2 ^3

endif
#-----
# Newton-Raphson + plot

e := n
if ((!3)) then read -p "edit nr parameters [y/n]? " e endif
if ((^e = y)) then xdmcc ce data/nrpar.data ; read -p "continue [y/n]? " c endif

func := gauss
read -p " gauss/raised cos [g/c]? " gc
if (( ^gc = c )) then func := rcos endif
args "===== NR ====="

read -p "percent emittance" percent

nr <<%
^func
^percent
data/nrpar.data
wire.data
wire.temp
ex.pdat
ey.pdat
%

catf wire.temp >> emit.data

mvf wire.temp wire.data -r

EFPLOT ^1 ^2 ^3

#-----
dlf ex.pdat ey.pdat ex.pbias cplot.in temp xi_file
#-----
# Use matrices from qm.data and sigmas from emit.data to find emittance

emittance <<%
qm.data
emit.data
emit.temp
%

catf emit.temp >> emit.data ; dlf emit.temp
```

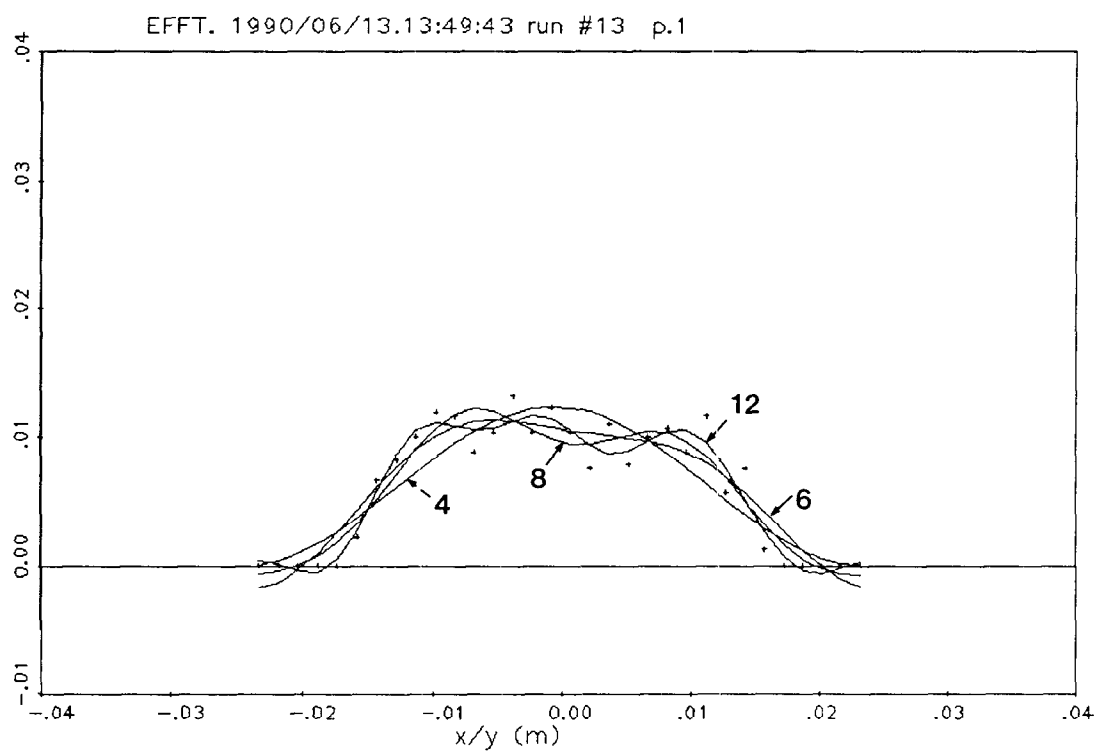


Fig. 1. FFT smoothing of 32 harp readings.

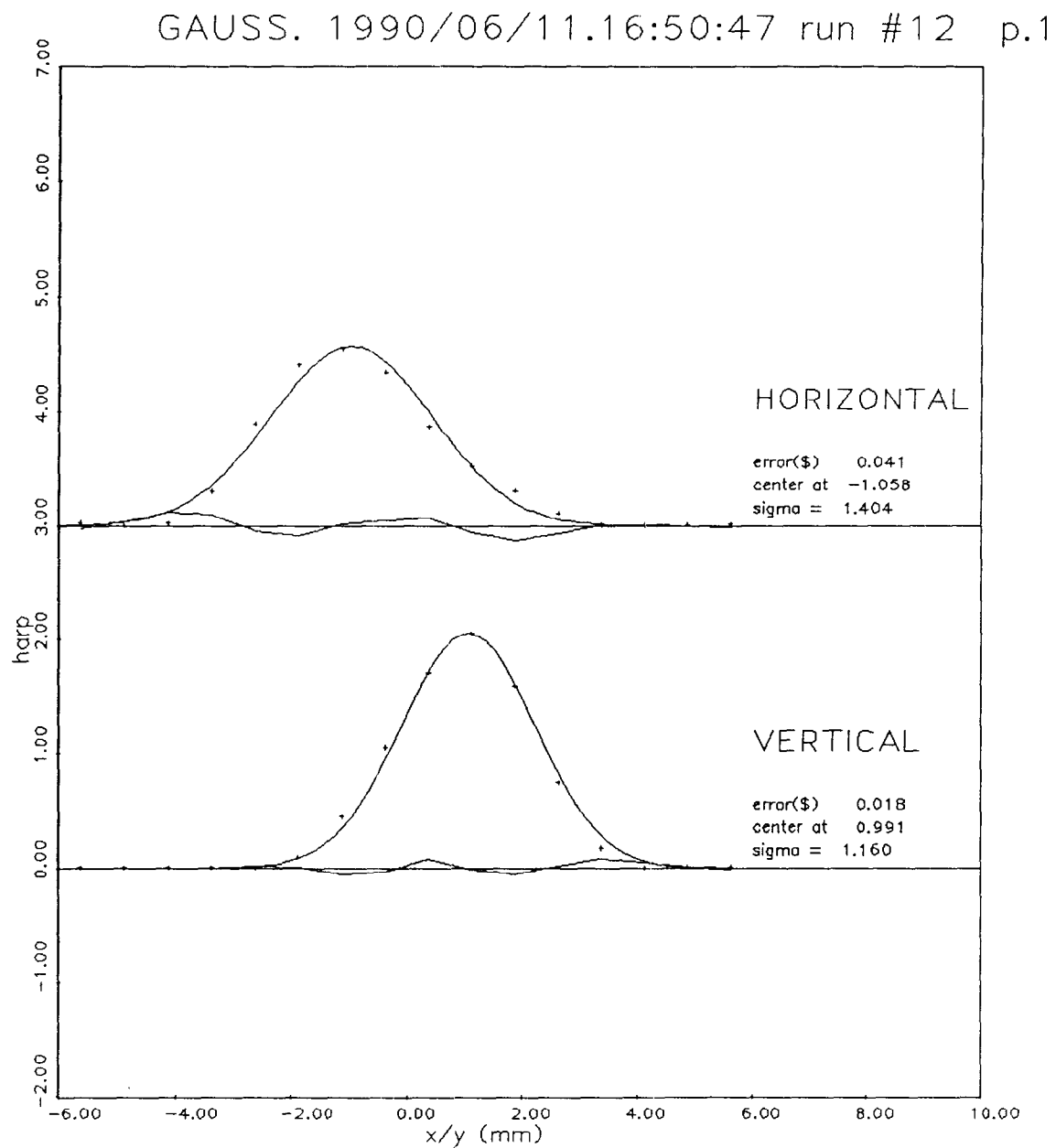


Fig. 2 a. Example of wire data fitting with a gaussian. "Nice data". The original points, the fitted curve and the error curve are represented.

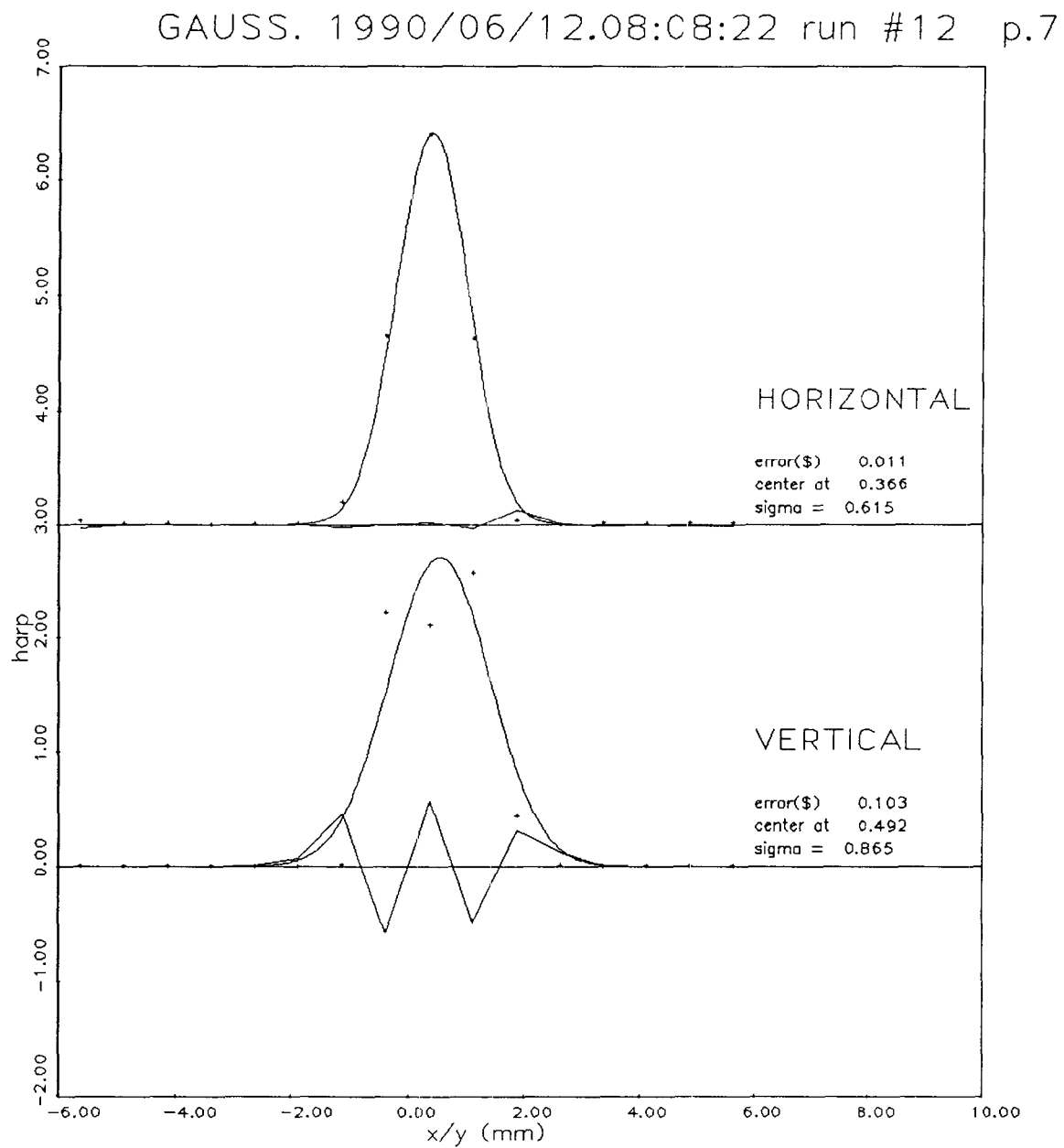


Fig. 2 b. Example of wire data fitting with a gaussian. "Not-so-nice data". The original points, the fitted curve and the error curve are represented.

RCOS. 1990/06/11.14:50:33 run #11 p.1

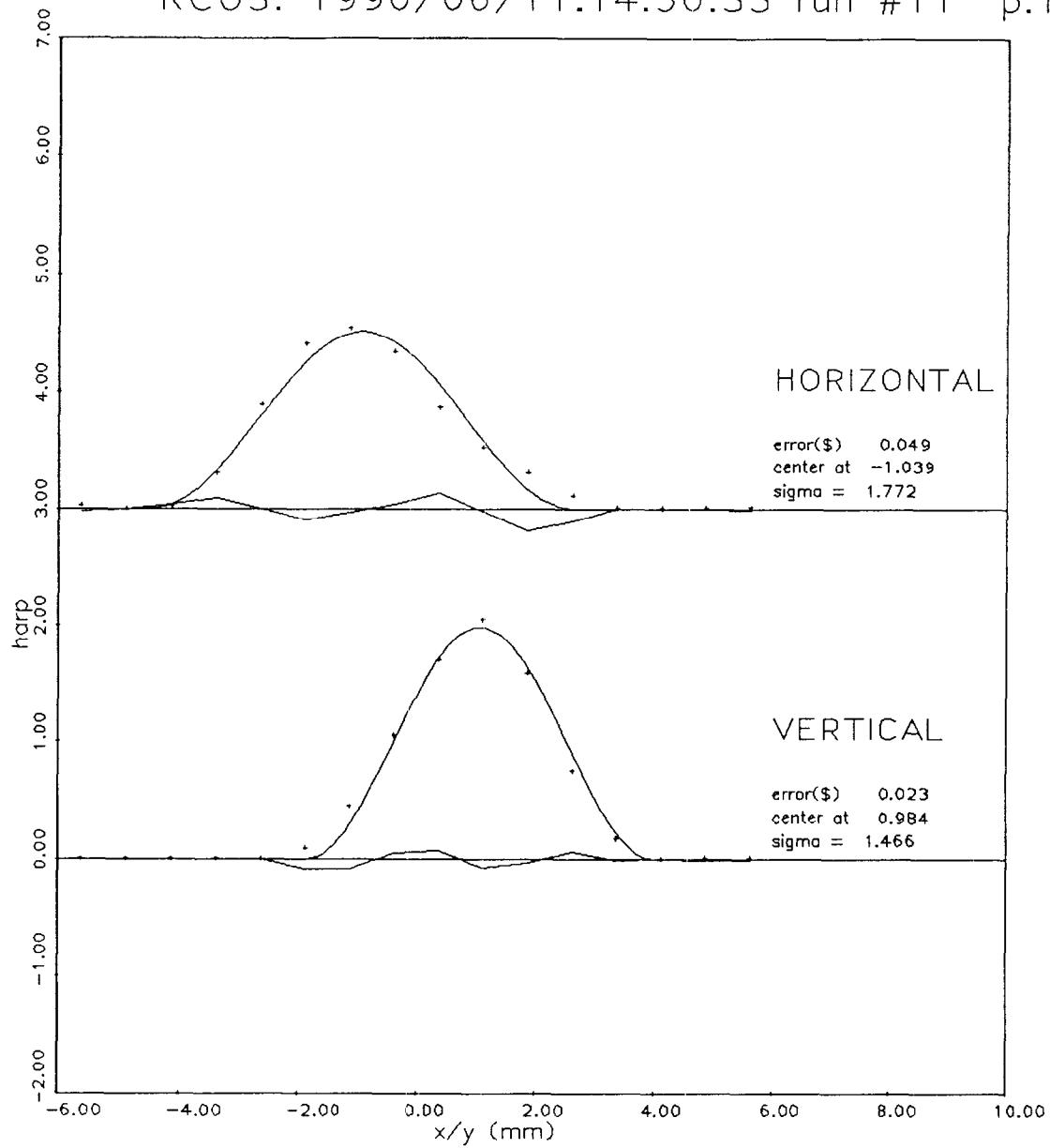


Fig. 3 a. Examples of wire data fitting with a raised cosine. Same "nice" data of Fig. 2a.
The original points, the fitted curve and the error curve are represented.

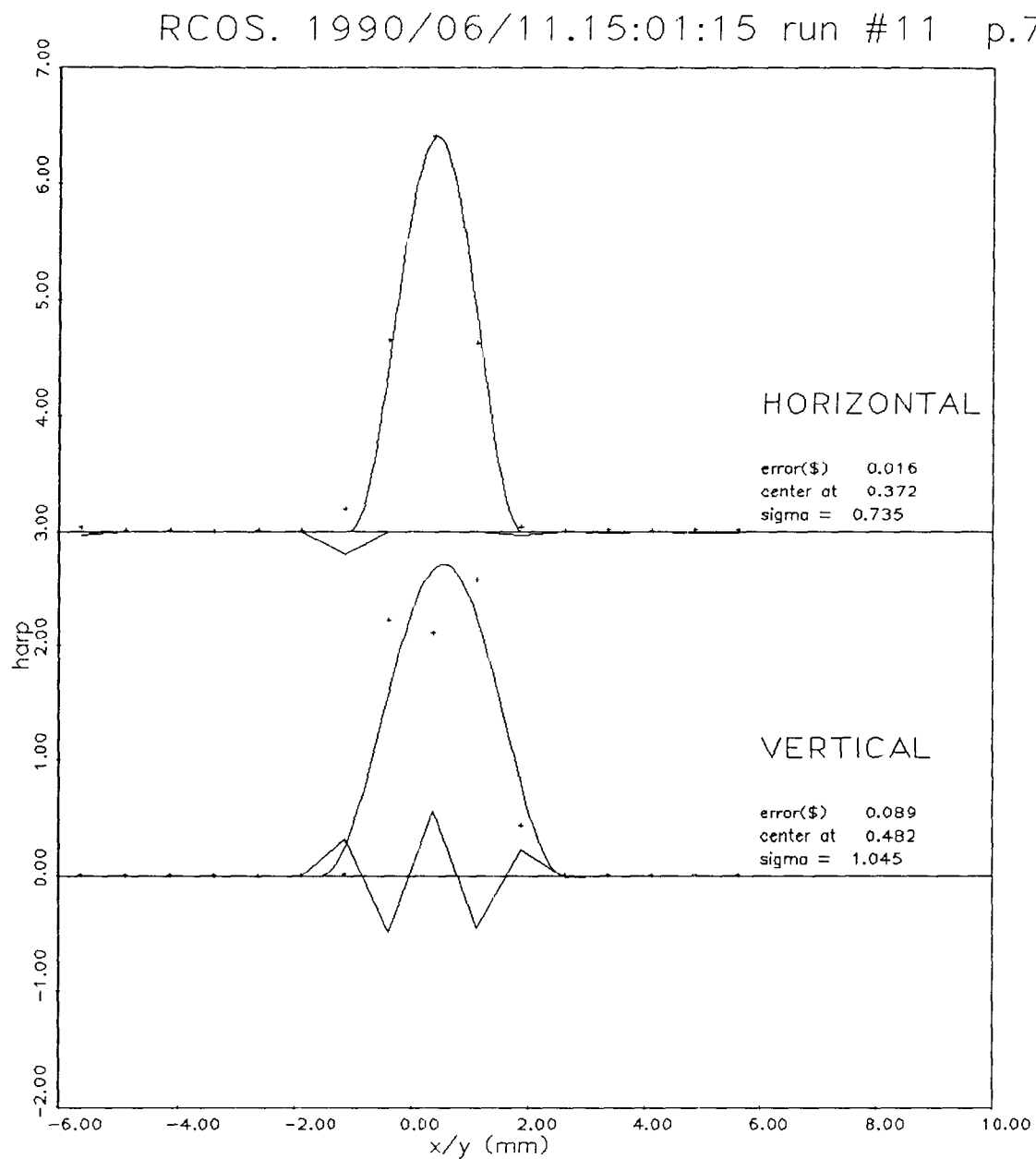


Fig. 3 b. Examples of wire data fitting with a raised cosine. Same "not-so-nice" data of Fig. 2b. The original points, the fitted curve and the error curve are represented.

EMIT. 1990/05/31.10:16:59 run #0 p.1

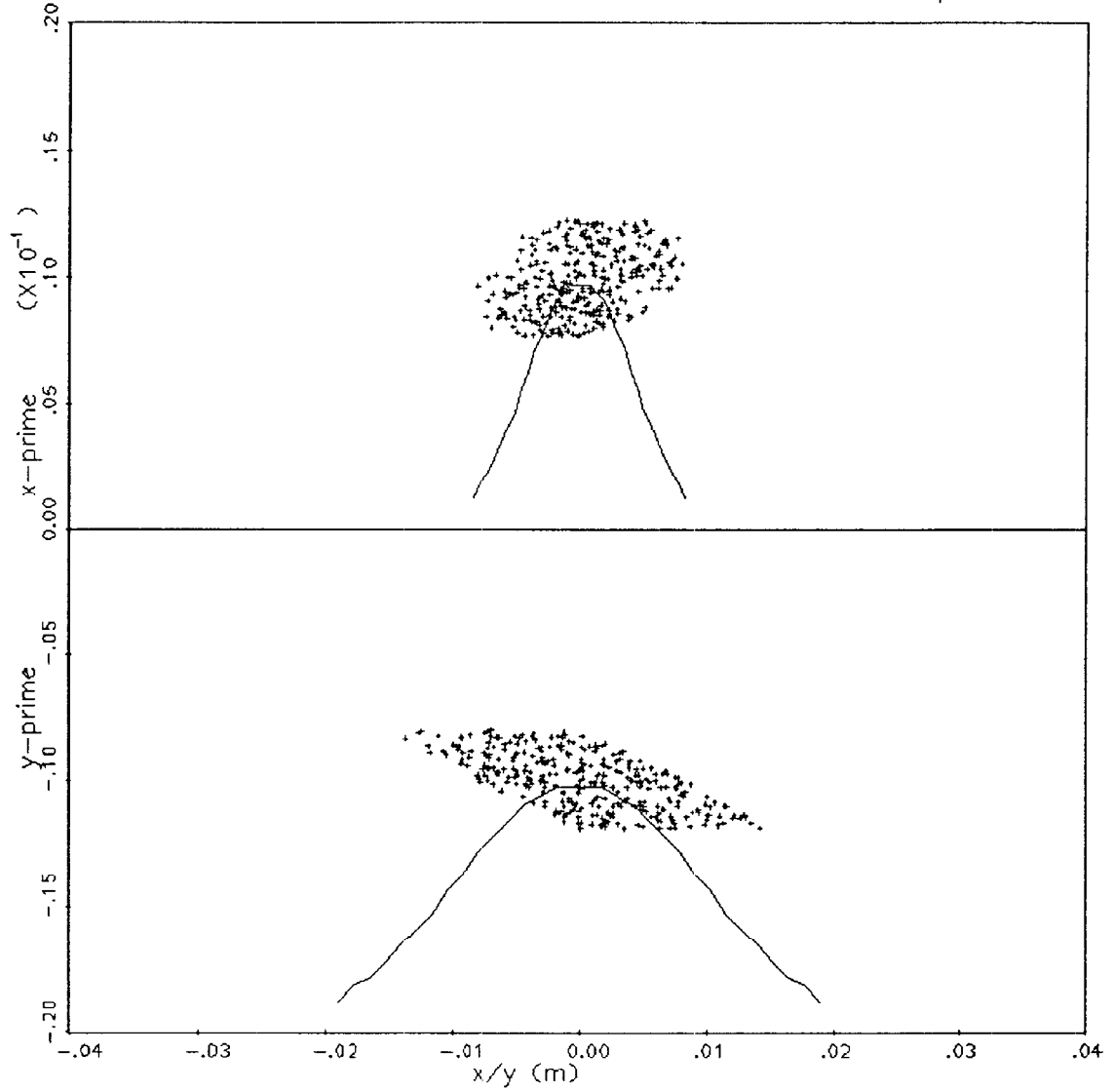


Fig. 4. A simulated beam is injected in the LTB line. 160 particles are in the clouds (horizontal, above, vertical, below) that top the mountains, representing, in arbitrary units, the beam profiles. The distribution uses 32 bins, both in horizontal and vertical.

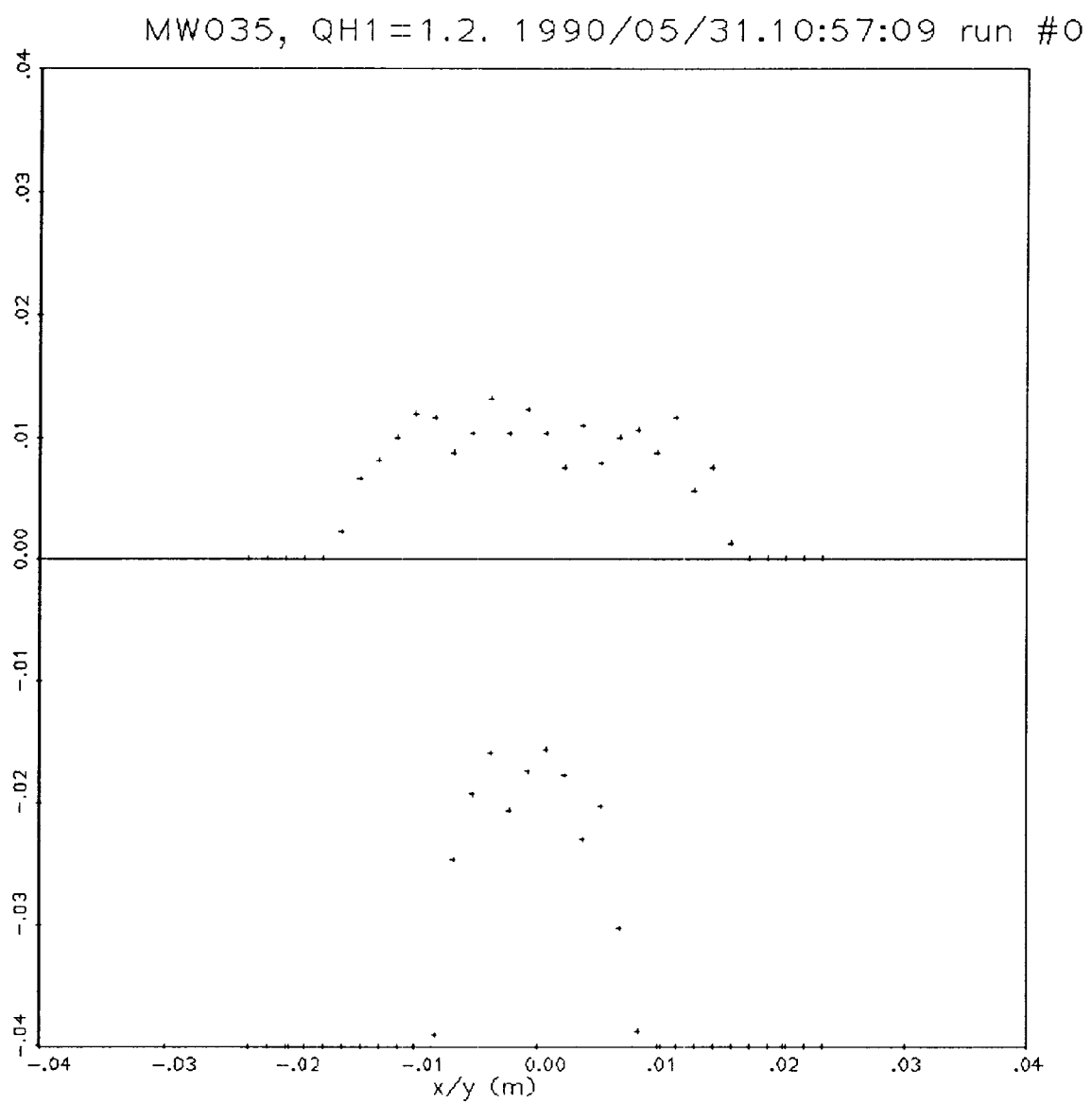


Fig. 5 . Examples of simulated readings in the LTB from the beam of Fig. 4. The harp has 32 wires.

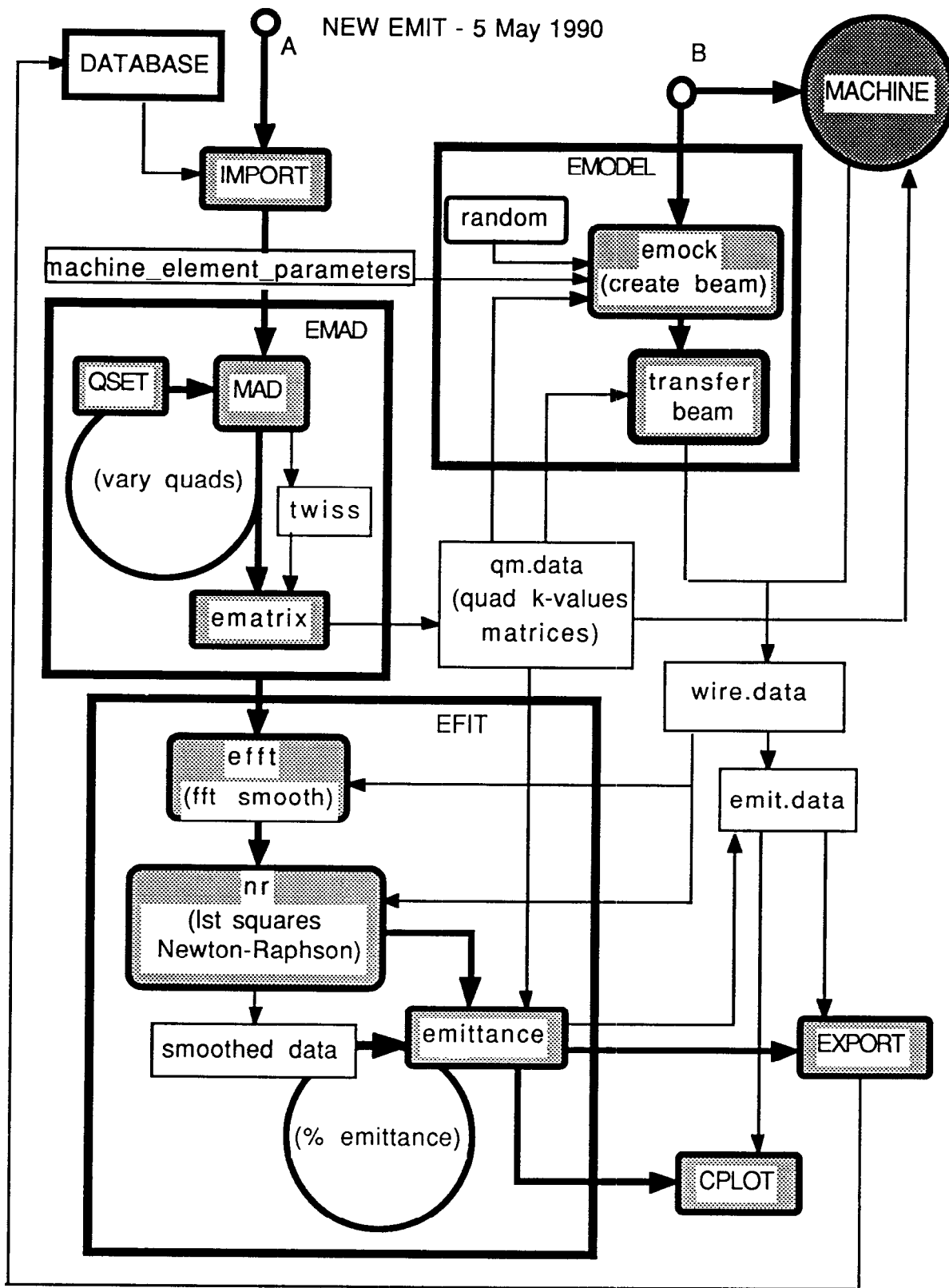


Fig.6. Flowchart of NEW EMIT

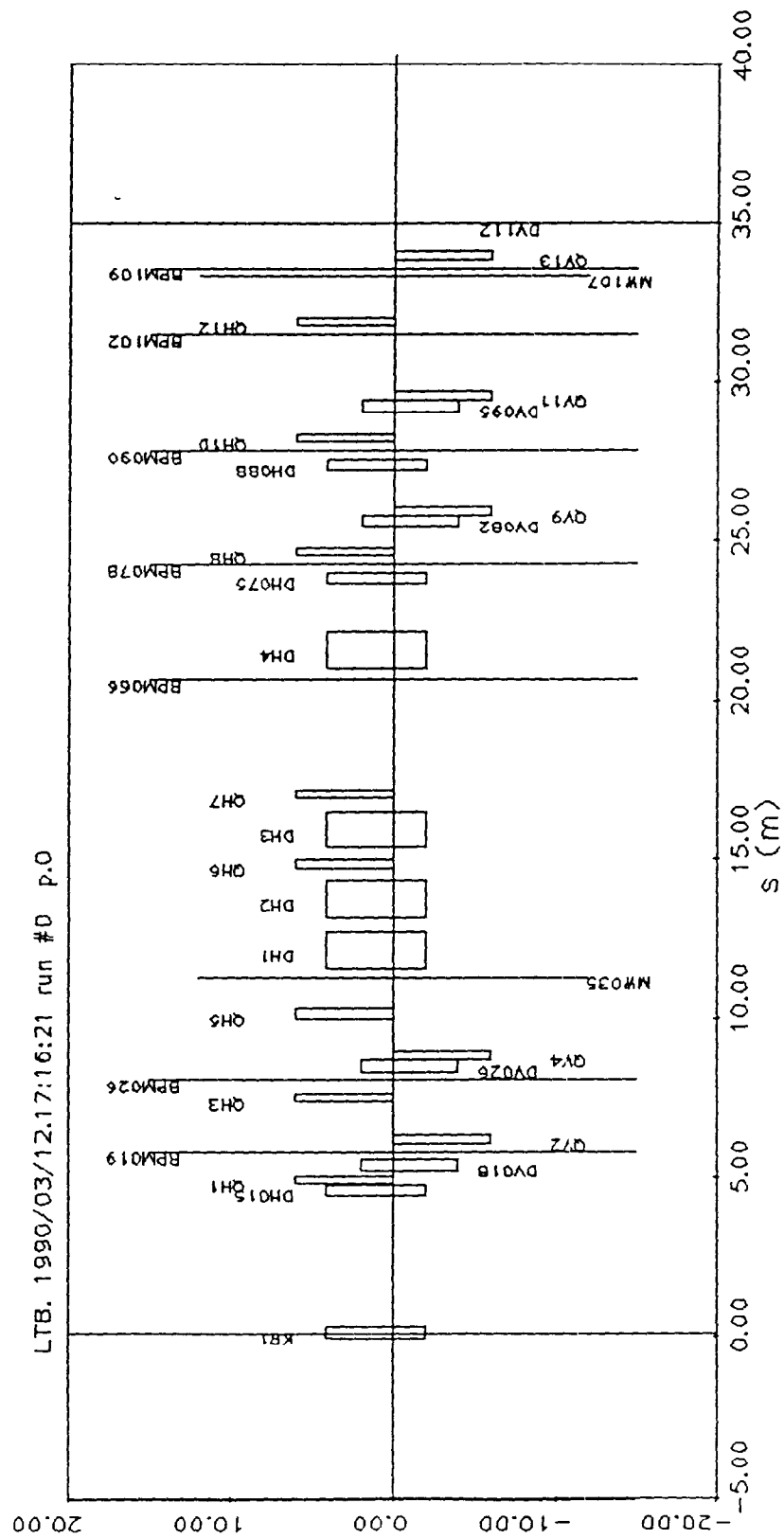


Fig. 7 . Layout of the LTB line. From⁶

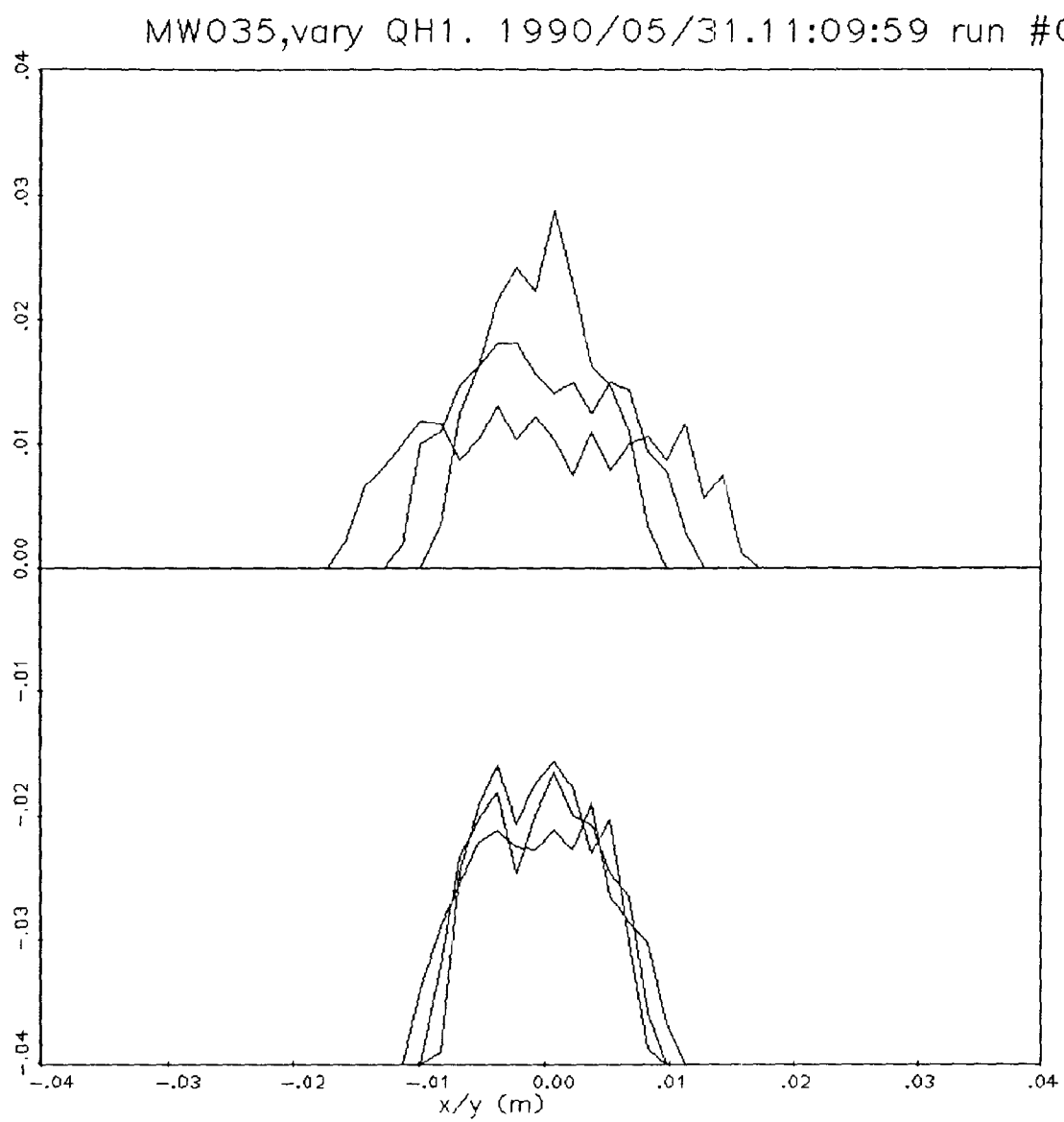


Fig. 8 a. Simulated readings at MW035 in the LTB line when QH1 is varied.

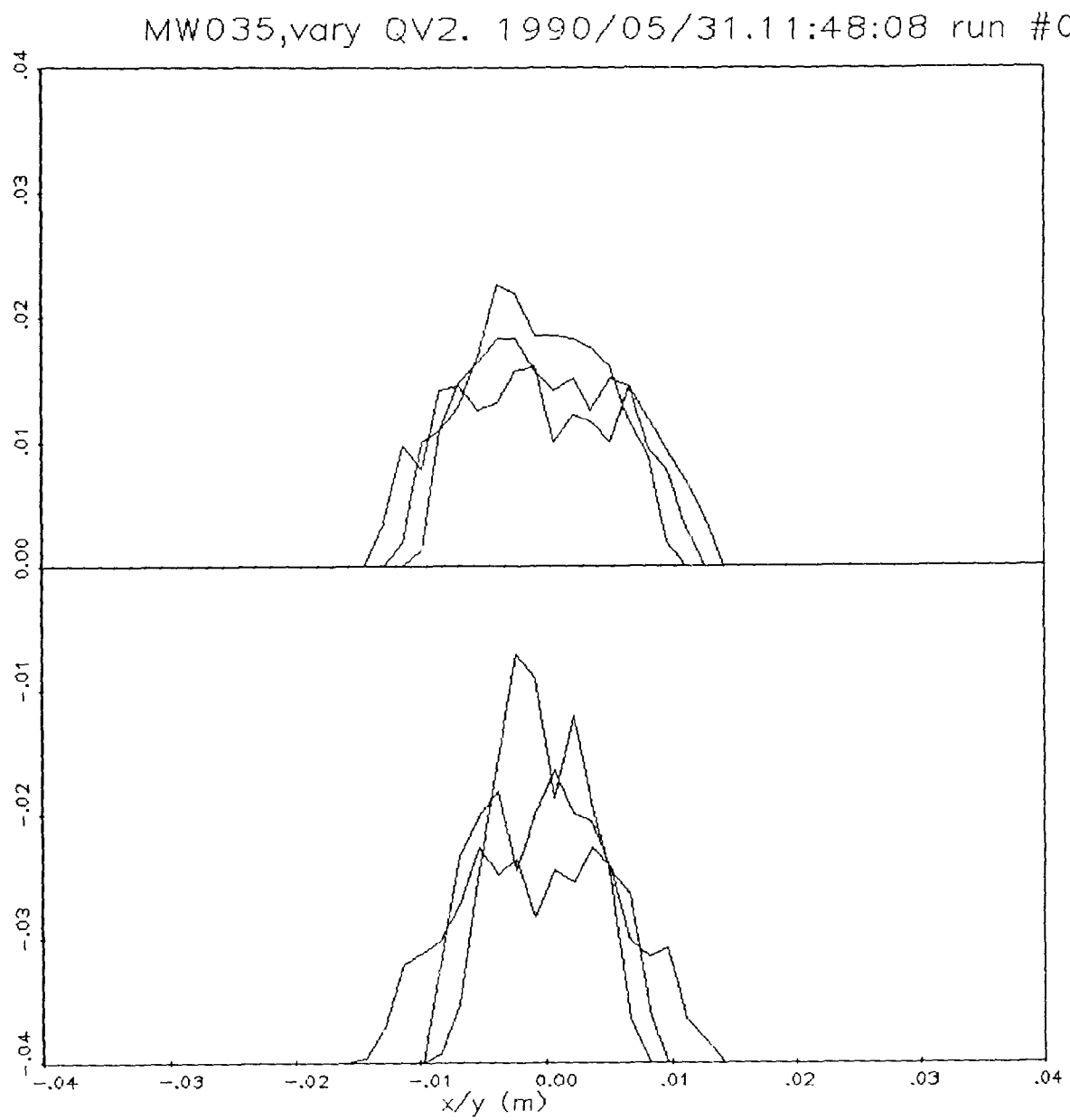


Fig. 8 b. Simulated readings at MW035 in the LTB line when QV2 is varied.

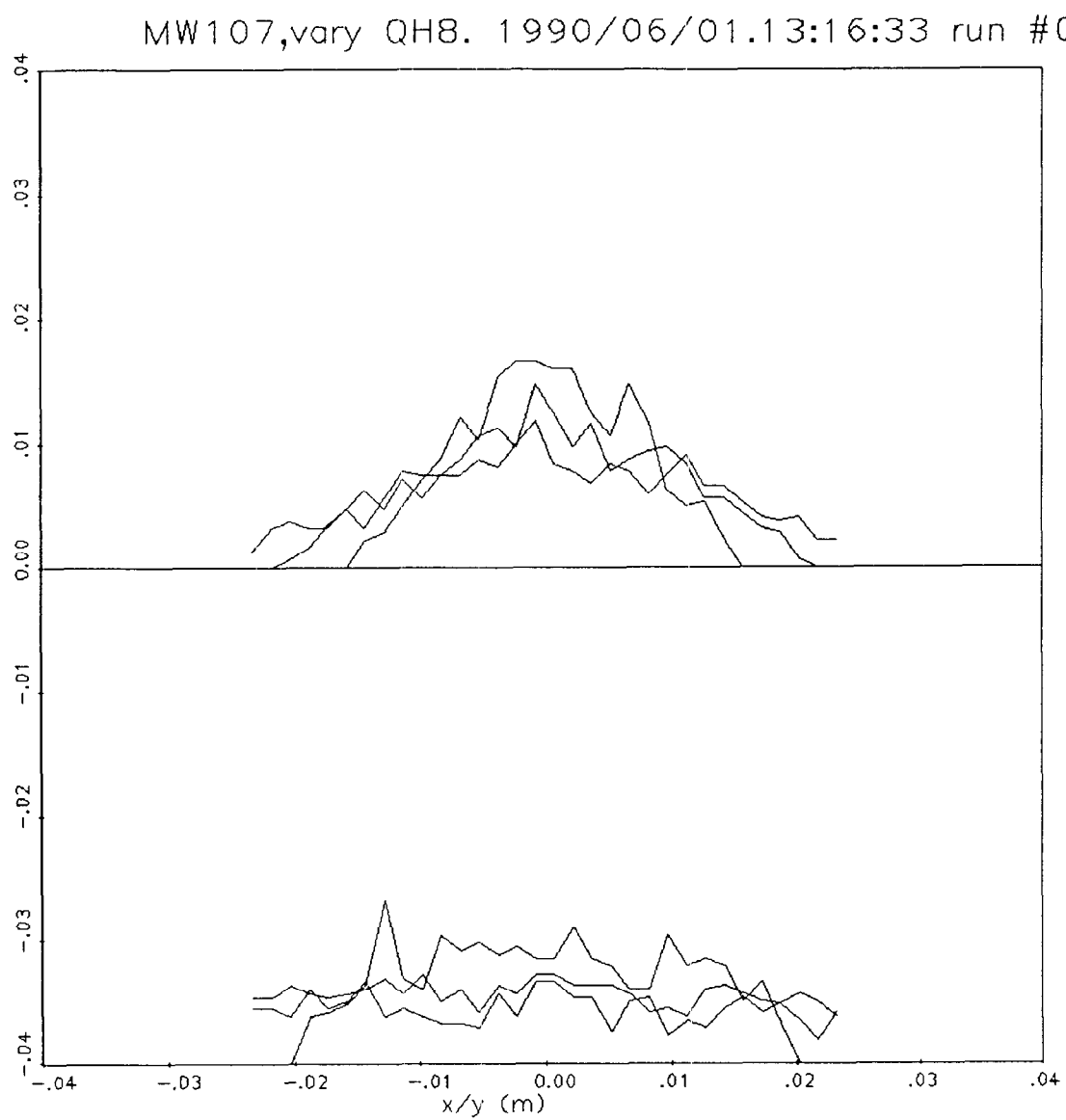


Fig. 9 a. Simulated readings at MW107 in the LTB line when QH8 is varied.

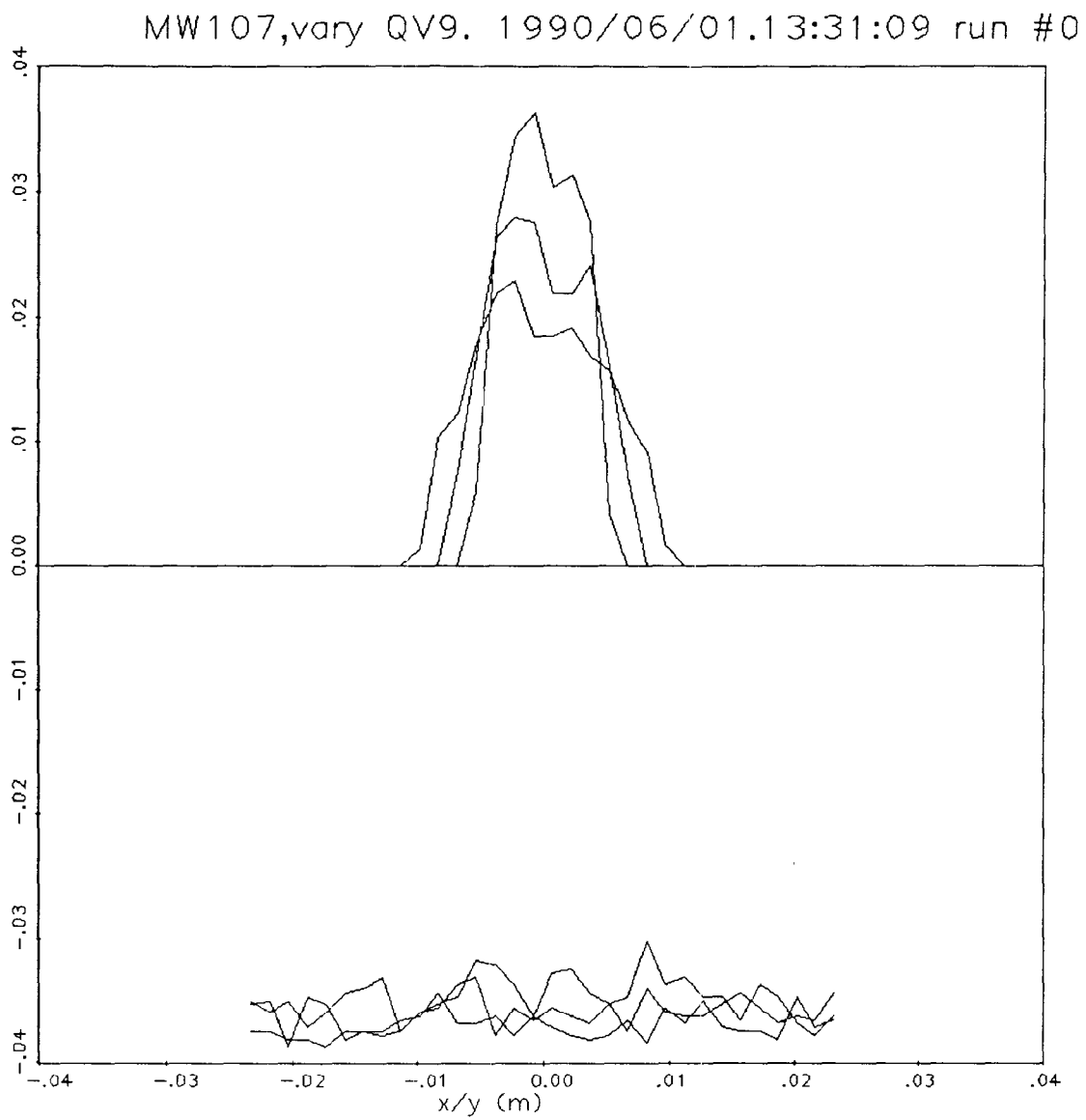


Fig. 9 b. Simulated readings at MW107 in the LTB line when QV9 is varied.

Appendix F. Fragment of the file qm.data for the LTB line. The "-2matrix" is the matrix of Eq. (2). The "-3matrix" is the matrix of Eq. (6).

//acn40d01/reality/luccio/pline/harp/qm.data

Wednesday, May 30, 1990 10:07:31 am (EDT)

file qm.data

run 5

quadrupole setpoints and transfer matrices between 0 and harps
from: machine_element_parameters

dataset 1

monitor BLI.MW035

k1(BLI.QH1) = 1.200000

k1(BLI.QV2) = -1.600000

k1(BLI.QH3) = 0.600000

k1(BLI.QV4) = -1.200000

k1(BLI.QH5) = 1.600000

x-2matrix

-0.2220498 6.758167

-0.8862328E-01 -1.806212

x-3matrix

0.4930612E-01 3.001299 45.67282

-0.1967878E-01 -0.1978619 12.20668

0.7854085E-02 -0.3201448 3.262402

y-2matrix

-0.1226975 3.276340

-0.2849072 -0.5423673

y-3matrix

0.1505468E-01 0.8039975 10.73440

-0.3495740E-01 -0.8669057 1.776980

0.8117211E-01 -0.3090487 0.2941623

dataset 2

monitor BLI.MW035

k1(BLI.QH1) = 1.400000

k1(BLI.QV2) = -1.600000

k1(BLI.QH3) = 0.600000

k1(BLI.QV4) = -1.200000

k1(BLI.QH5) = 1.600000

x-2matrix

-0.5801424 4.738007

-0.3022733E-01 -1.476849

x-3matrix

0.3365653 5.497438 22.44871

-0.1753616E-01 0.7135653 6.997320

0.9136916E-03 -0.8928239E-01 2.181082

y-2matrix

0.6919499E-01 4.358696

-0.2334244 -0.2518391

y-3matrix

0.4787948E-02 -0.6032000 18.99824

0.1615180E-01 -1.034852 1.097690

0.5448693E-01 -0.1175708 0.6342293E-01

dataset 3

monitor BLI.MW035

k1(BLI.QH1) = 1.600000

k1(BLI.QV2) = -1.600000

k1(BLI.QH3) = 0.600000

k1(BLI.QV4) = -1.200000

k1(BLI.QH5) = 1.600000

x-2matrix

-0.9360963 2.729705

0.2782324E-01 -1.149400

x-3matrix

0.3365653 5.497438 22.44871

Appendix G. Fragment of emit.data for the LTB line with the 50% emittance calculated for the simulated beam and up to 20 combinations of 6 datasets by 3.

//acn40d01/reality/luccio/pline/harp/emit.data

Friday, June 15, 1990 5:09:00 pm (EDT)
file emit.data
run 15
result of wire data analysis

dataset 1

wires(x): 32

wires(y): 32

measurement: BLI.MW035

-2.3250001E-02 1.1099691E-03

-2.1749999E-02 1.5092628E-03

-2.0250000E-02 2.0092933E-03

-1.8750001E-02 2.6190632E-03

-1.7250000E-02 3.3425137E-03

-1.5750000E-02 4.1766667E-03

-1.4250000E-02 5.0000000E-03

-1.2750000E-02 5.8333333E-03

-1.1250000E-02 6.6666667E-03

-1.0000000E-02 7.5000000E-03

-8.7500000E-03 8.3333333E-03

-7.5000000E-03 9.1666667E-03

-6.2500000E-03 1.0000000E-02

-5.0000000E-03 1.0833333E-02

-3.7500000E-03 1.1666667E-02

-2.5000000E-03 1.2500000E-02

-1.2500000E-03 1.3333333E-02

-0.0000000E+00 1.4166667E-02

0.0000000E+00 1.5000000E-02

0.0000000E+00 1.5833333E-02

0.0000000E+00 1.6666667E-02

0.0000000E+00 1.7500000E-02

0.0000000E+00 1.8333333E-02

0.0000000E+00 1.9166667E-02

0.0000000E+00 2.0000000E-02

0.0000000E+00 2.0833333E-02

0.0000000E+00 2.1666667E-02

0.0000000E+00 2.2500000E-02

0.0000000E+00 2.3333333E-02

0.0000000E+00 2.4166667E-02

0.0000000E+00 2.5000000E-02

0.0000000E+00 2.5833333E-02

0.0000000E+00 2.6666667E-02

0.0000000E+00 2.7500000E-02

0.0000000E+00 2.8333333E-02

0.0000000E+00 2.9166667E-02

0.0000000E+00 3.0000000E-02

0.0000000E+00 3.0833333E-02

0.0000000E+00 3.1666667E-02

0.0000000E+00 3.2500000E-02

0.0000000E+00 3.3333333E-02

0.0000000E+00 3.4166667E-02

0.0000000E+00 3.5000000E-02

0.0000000E+00 3.5833333E-02

0.0000000E+00 3.6666667E-02

0.0000000E+00 3.7500000E-02

0.0000000E+00 3.8333333E-02

0.0000000E+00 3.9166667E-02

0.0000000E+00 4.0000000E-02

0.0000000E+00 4.0833333E-02

0.0000000E+00 4.1666667E-02

0.0000000E+00 4.2500000E-02

0.0000000E+00 4.3333333E-02

0.0000000E+00 4.4166667E-02

0.0000000E+00 4.5000000E-02

0.0000000E+00 4.5833333E-02

0.0000000E+00 4.6666667E-02

***** HORIZONTAL *****

Combination(1): 1 2 3
emit= 0.639E-05, beta= 3.10 , alfa=-0.463
Combination(2): 1 2 4
emit= 0.661E-05, beta= 3.20 , alfa=-0.463
Combination(3): 1 3 4
emit= 0.634E-05, beta= 3.06 , alfa=-0.454
Combination(4): 1 3 5
emit= 0.639E-05, beta= 3.10 , alfa=-0.463
Combination(5): 1 4 5
emit= 0.661E-05, beta= 3.20 , alfa=-0.463
Combination(6): 2 3 4
emit= 0.622E-05, beta= 3.00 , alfa=-0.434

Average over 6 combinations
emit_x= 0.643E-05+- 0.157E-06
beta_x= 3.11 +- 0.774E-01
alfa_x=-0.457 +- 0.118E-01

***** VERTICAL *****

Combination(1): 1 2 3
emit= 0.548E-05, beta= 7.03 , alfa= 1.08
Combination(2): 1 2 4
emit= 0.537E-05, beta= 7.05 , alfa= 1.10
Combination(3): 1 2 6
emit= 0.481E-05, beta= 7.21 , alfa= 1.24
Combination(4): 1 3 4
emit= 0.520E-05, beta= 7.03 , alfa= 1.13
Combination(5): 1 3 5
emit= 0.548E-05, beta= 7.03 , alfa= 1.08
Combination(6): 1 3 6
emit= 0.498E-05, beta= 7.06 , alfa= 1.18

Average over 6 combinations
emit_y= 0.522E-05+- 0.276E-06
beta_y= 7.07 +- 0.696E-01
alfa_y= 1.14 +- 0.635E-01